

TECHNICAL OVERVIEW

ESIG4000

**The New
Image Generator
Architecture
For The 90s**

EVANS & SUTHERLAND

ESIG-4000

THE NEW IMAGE GENERATOR ARCHITECTURE FOR THE 90s

Evans & Sutherland introduces the ESIG-4000, a revolutionary new approach to computer image generation. This machine establishes a new system architecture which separates the processing of terrain and features in both hardware and modeling tools. These advancements offer for the first time in one machine unparalleled photo-realistic visual fidelity and rapid database generation, making it the ideal choice for mission rehearsal as well as low altitude, high-speed, fixed wing and helicopter nap-of-earth pilot training, and vehicle engineering simulation.

True Mission Rehearsal Performance

For several years simulation users have sought ways to reduce database generation times. Initially the motive was simply economics – the faster a database could be made, the less expensive it was. More recently the elusive rapid-database-generation goal has been aggressively championed by the mission rehearsal (MR) community, where databases often must be produced from source materials (photos, Project 2851 and DMA data, maps, etc.) in as few as 48 hours.

Mission rehearsal, primarily encompasses three very difficult visual requirements:

- Very high fidelity to the real world
- Large area databases
- Extremely short database generation times

Unfortunately, with conventional image generator architectures, solutions to these three requirements are almost unavoidably mutually exclusive. Anything done to improve one makes the others more difficult.

With the new ESIG-4000 image generator, an entirely different architectural approach provides image generation techniques which finally make it possible to address these difficult MR requirements, while preserving the high scene quality which has been characteristic of E&S image generators for many years.

Key ESIG-4000 Features For Advanced Simulation Requirements

The remainder of this document discusses several of the new ESIG-4000 features which are especially crucial to addressing the mission rehearsal/rapid database creation challenge:

- Separate processing of features and terrain in both modeling tools and hardware
- "On-the-fly" combining or "conforming" of features to terrain in the hardware
- Range-Buffer for generalized visual priority, including inter-penetration of surfaces
- Full-color texture, including Global Texture for photo-textured terrain to under 1-meter size
- "Betweening" for smooth, continuous LOD evolution of terrain and features
- Database compression, higher terrain accuracy, and faster flight via run-time generation of terrain skin from a grid of elevation values
- Robust Non-Linear Image Mapping for advanced display systems
- Enhanced instancing for improved feature correlation with source data

Simplifying Database Generation: Separation of Terrain and Features

One of the most dramatic advancements in the architectures of CIG systems is the new capability of the ESIG-4000 to *separately* process the features and the terrain comprising a complete database. The implications of this capability are far reaching, including both the database modeling phase and the run-time capabilities of the system. A brief overview of this powerful new capability is presented next.

Features and Terrain Created Separately in the Modeling Tools

One of the most time-consuming aspects (for both computers and people) with current database generation procedures is the complex interaction between the terrain skin and the 3-D features which are to be placed upon that terrain. The figure on the opposite page illustrates a powerful new ESIG-4000 capability— separation of terrain and features. An ESIG-4000 "database" actually consists of several separate databases, each created and stored separately and independently. This permits *parallel* development of these databases, providing a significant increase in database throughput simply by this feature alone. Notice the very important point that **any database changes in 3-D features may be made without any corresponding changes being required in the terrain, and vice versa.** This separation greatly reduces the complexity of incremental database changes.

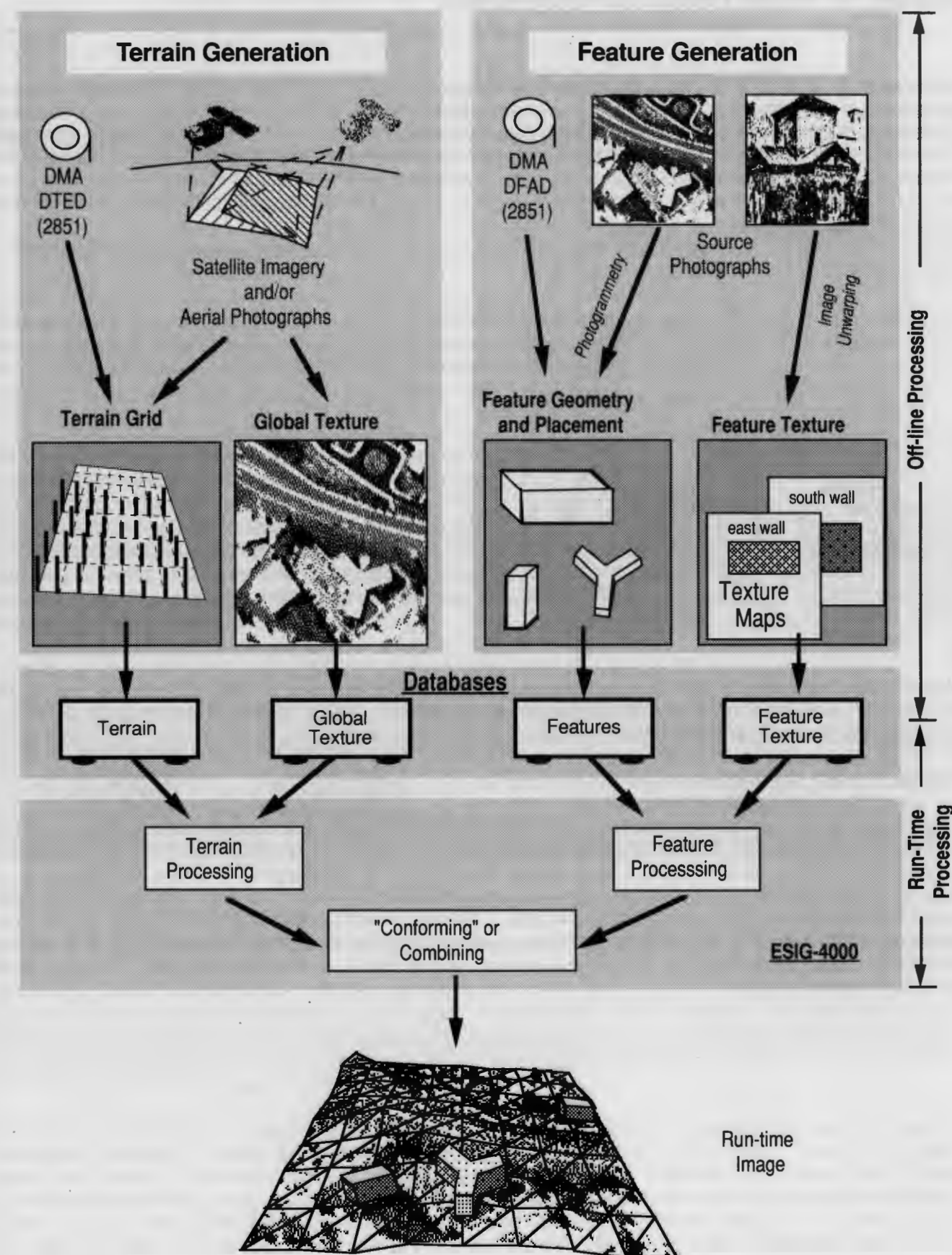
Features and Terrain Processed Separately in the Hardware

It is also important to note that there are two distinct aspects to this separation. First, terrain and feature databases are *created* separately (by the modeling tools) as mentioned above. Secondly, the terrain and feature databases are *processed* separately in the image generator hardware, and then combined *in the image generator hardware*. This allows the characteristics of terrain and features to be separately optimized at run time, on a dynamic basis (responding to current IG polygon load conditions, etc.), before they are finally combined in the latter stages of the image generator.

This extremely powerful capability should not be confused with a simple modeling-tool scheme which merely creates the features and terrain separately, and then combines these together *in the modeling tools* into a single executable database. Combining terrain and features in the modeling tools places a heavy burden on the modeler and the modeling tools (equating to long database design, compile and/or link times, and heavy storage requirements), and doesn't realize the polygon economies and database creation efficiencies available with the ESIG-4000.

In order to fully appreciate the power of this "separation" feature, the next section reviews some of the undesirable effects of the interaction between features and terrain. Unfortunately these tedious interactions are unavoidable with current image generation architectures. However, with the hardware separation of terrain and features inherent in the ESIG-4000 architecture, these limitations and constraints are simply *eliminated*.

Separation of Terrain and Features in the ESIG-4000



Terrain/Feature Interaction: Enemy of Rapid Database Generation

Generation of visual databases generally consists of two major partitions — generation of the terrain skin, and generation of the features which fit on the terrain skin. Until the ESIG-4000, one of the most time consuming tasks in any database creation has always been properly combining these two partitions into a single database. This terrain-feature combining operation generally requires detailed examination of both the terrain and the feature databases many times during the database creation process. Worse, these examinations often need help from the database modeler as conflicting situations arise. This section mentions a few of these time-consuming problems.

Terrain/Feature Fit Problems

Previous to the ESIG-4000, it was not possible to consider the creation of a feature without simultaneously considering the terrain upon which the feature was to be placed. This made it necessary to know in advance both the terrain slope and the terrain elevation at the exact location where a feature was to be placed.



For example, note in this illustration that the house fits properly on flat terrain, but is obviously not suitable for placement on sloping terrain. This often leads to the unfortunate necessity to create a large number of variations of each feature type, for each level of detail, which is very time-consuming, and wasteful of precious disk and IG memory storage space.

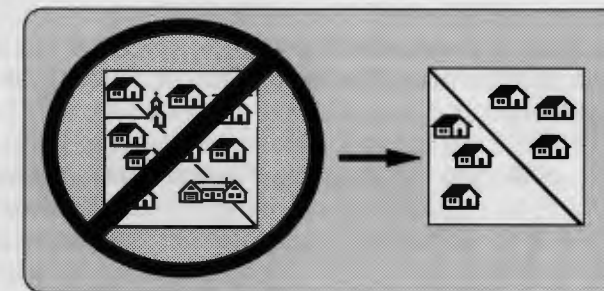
Note also that this terrain/feature fit problem, in addition to making creation and storage of features much more difficult, also **requires that the terrain database be examined for each individual feature to be placed**—again a very labor- and compute-intensive operation.

Visual Priority Conflicts

The most common high-performance visual system architecture for over a decade has been based on the concept of cellular priority. This architecture requires that any feature in the database cannot straddle a terrain polygon boundary, or proper visual priority cannot be assured. In addition, if the number of features to be included within a terrain triangle area is larger than just a few, the calculation of proper visual priority becomes very complex for computer-automated procedure. This often leads to labor-intensive manual intervention, resulting in alteration of the terrain polygon boundaries, or relocation of features away from their correct positions. Often, the expedient, but undesirable solution is simply to eliminate features, reducing visual database feature density.

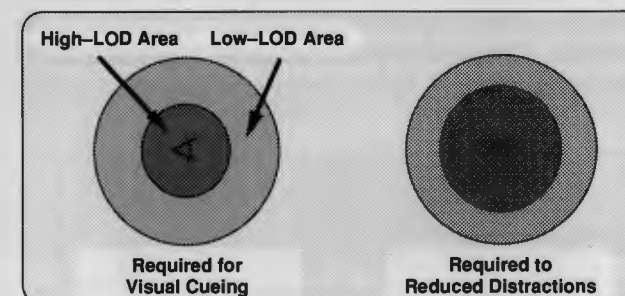
The important point is that with current architectures **it is necessary to examine and/or modify the terrain or feature for each feature which is placed**. This process is a very compute-intensive operation, often requiring human assistance to help sort out potential conflicts. The resultant difficulty of placing features on the terrain drastically increases the complexity of database creation.

In this illustration, after repeatedly examining the terrain and feature interaction, several 3-D features were discarded to avoid conflicts, *resulting in decreased feature density*. Each of the original ten features required the terrain database to be checked for possible error conditions. These constraints also often lead to a repetitive or “patterned” appearance in placement of database features.



Level-of-Detail Constraints

Prior to ESIG-4000, there were some necessary and generally undesirable interactions between terrain, features, and level-of-detail (LOD) transitioning. Previous systems, incorporating discrete geometry changes between adjacent LODs, must locate LOD transitions at long distances from the eyepoint to “hide” these visually-distracting transitions. This means that higher polygon LODs must be preserved to greater distances, leading to increased polygon load.



This figure shows two bird's-eye views. The left view shows the range from the eyepoint where the pilot's visual “observability” would naturally allow transitioning to a lower polygon LOD. In contrast, the right side indicates the larger high-LOD area generally required with discrete geometry discontinuities between levels of detail.

A significant advancement in LOD transitioning was introduced in the E&S CT5-A. This capability—fade level of detail, or FLOD—permits transparency “fading” between two discrete versions of terrain or feature geometry. While this technique provides improvement, it does require that two versions of geometry be simultaneously processed, leading to some polygon processing penalty. In addition, since the two versions being blended necessarily have different geometry, the LOD transition still needs to be located at increased distance from the eyepoint to minimize distracting effects—again leading to retention of more high-LOD information than is really required to support visual cueing.

The important point is that “modeling-time” placement of features on terrain (by the modeling tools) into a single executable database virtually precludes any possibility that terrain LOD can “evolve” smoothly or continuously with range. If the terrain *could* be made to evolve continuously, the features on the terrain would have to move correspondingly to avoid “fit” problems. This, of course, is not possible when the terrain/feature fitting is performed for discrete terrain LODs at database compile time. Thus, transition ranges must continue to be artificially increased to hide distracting transition effects, wasting valuable polygon processing capacity.

Separation of Terrain and Features: Overview

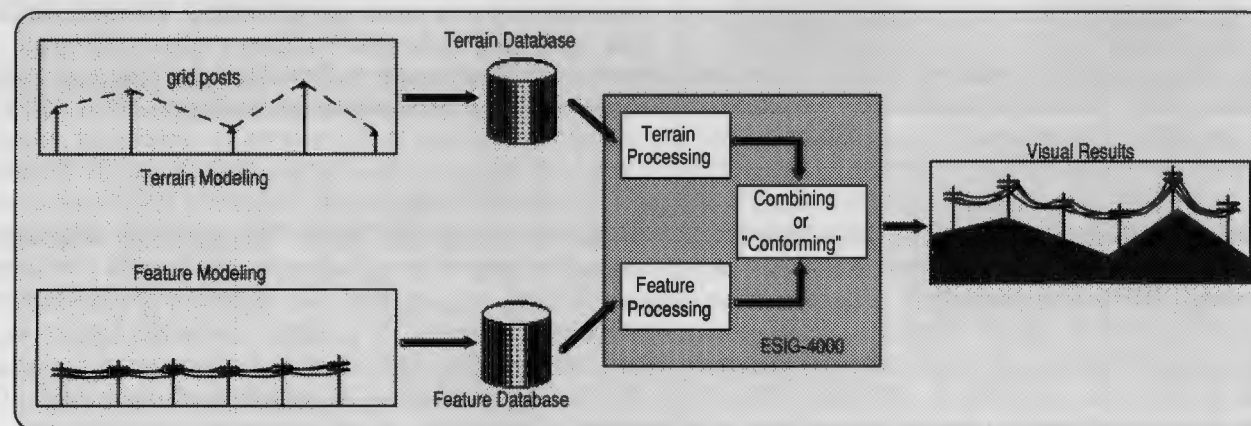
ESIG-4000 Features Eliminate Database Constraints

Several architectural features in the ESIG-4000 specifically address these difficult problems, making the rapid generation of visual databases a practical reality. These enabling features are briefly described in the next sections of this document.

The new ESIG-4000 "Separation of Terrain and Features" capability is the single most important feature in addressing the demanding mission rehearsal requirements mentioned in the introduction of this document. In order to accomplish this "separation" technique, the ESIG-4000 architecture incorporates the following general capabilities:

- **Conforming:** Features are combined with the terrain in the hardware, using a process referred to as "conforming." The hardware determines what the terrain geometry is like "under" the feature, and positions the feature properly.
- **R-Buffer:** A range buffer provides correct visual occultation for all database features, moving or static, and facilitates proper real time "trimming" of features where they meet the terrain.
- **Betweening:** Level of Detail evolves gradually and continuously for both terrain and features.
- **LOD Processing:** Terrain and Features are processed independently, allowing each to have its own optimal LOD management.

The following illustration depicts the separation process, and emphasizes the point that the terrain and features are modeled separately, stored separately, processed separately in the hardware, and then finally combined into a composite visual image. Changes can be made to either of the two databases without the necessity to "check" or modify the other database. Not only can each of these two separate databases be made much faster, but both can be developed simultaneously in parallel efforts. This not only makes large, rapid database creation possible, but also streamlines incremental database changes.



The ability to separately process terrain and features relies on several fundamental elements which are briefly described next. Each element provides an essential part of this powerful capability, and without any one element, the full potential for rapid generation of high-fidelity databases cannot be achieved.

Separate LOD Management for Terrain and Features

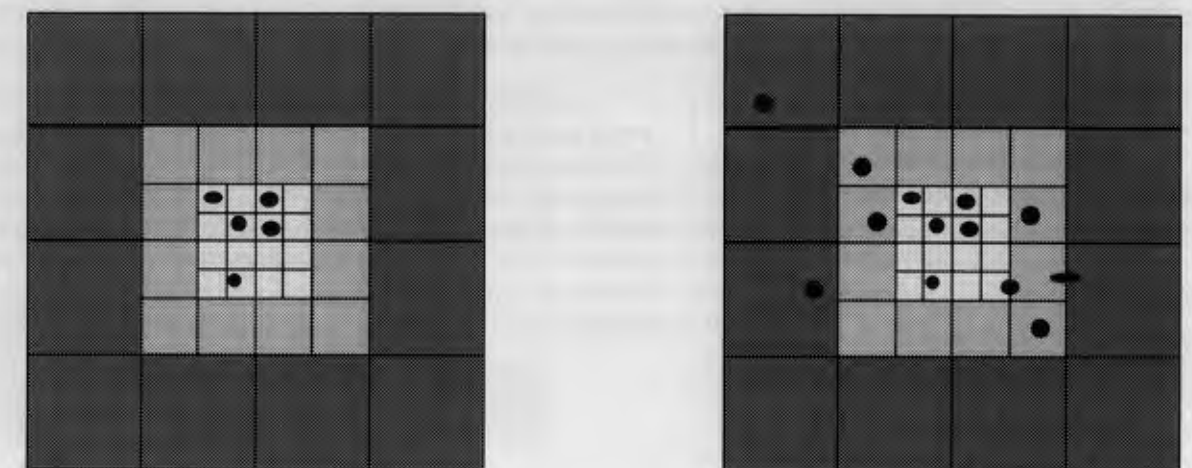
Separate LOD Processing for Terrain and Features

The primary purpose of any level-of-detail (LOD) management scheme is to provide the highest possible polygon density in the area where the pilot is focusing his attention. For most applications this means placing the highest level of detail close to the pilot. Since it takes fewer polygons per unit area to achieve the same visually-apparent detail at a longer viewing distance, polygon LOD is decreased aggressively with increasing distance, in a manner such that no visually distracting LOD changes are made.

It is often desirable to make different LOD decisions for the terrain, and for the features which are on that terrain. For example, certain waypoints or tactical targets may need to be maintained to high LOD out to a long distance, say 5 km., so that they can be properly identified using a high magnification sensor. Prior to the ESIG-4000, since the features and terrain were part of the same database, they shared the same LOD decisions. This naturally leads to difficult polygon capacity tradeoffs, generally requiring that the overall database be "thinned out" to allow these 3-D features to be preserved at high LOD. This of course affects the number of polygons which can be allocated to the terrain skin, resulting in degraded terrain accuracy.

With ESIG-4000, since terrain and features are modeled separately in the database tools, and processed separately in the image generator, each can make its own decisions about LOD transition ranges. High-LOD features can be placed on low-LOD terrain if desired, with a visually-proper combination resulting. Also, some 3-D features can have short LOD transition ranges, while other more-significant features have long transition ranges.

This concept is illustrated in the following figure. The small squares represent high LOD terrain close to the eyepoint, with decreasing LOD (larger terrain squares) farther from the eyepoint. On the left is the case where features are only allowed on the high-LOD terrain, with the resultant sparsity of 3-D features. On the right is the case where 3-D features can be placed on terrain of any LOD, allowing important features, such as targets and waypoints to be preserved to much longer distances. Also note that features are placed without regard to the location of terrain facet boundaries, allowing more accurate placement/correlation, and higher feature density because of retention of more features.



Combining Features and Terrain: "Conforming"

An essential element of the separation of terrain and features is the ability to combine them together *in the hardware*. This process is referred to as "conforming" the features to the terrain. Accomplishing this difficult task in the image generator hardware has a profound effect on the complexity of "populating" the terrain skin with features. The task of the modeler/modeling tools is reduced to simply placing the features at the desired locations, **as if they were being placed on an infinite flat plane**. The need to deal with varying terrain topography for feature placement is simply eliminated.

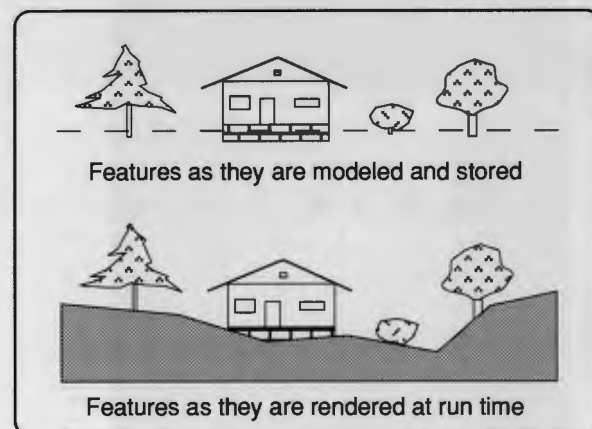
The conforming process involves determining, at run time, which piece(s) of terrain are "under" a feature, and then performing operations to place the feature at the correct elevation, while at the same time making it fit properly on the terrain. In addition, the feature may undergo "clipping" operations, if appropriate, to complete the conforming function. The R-Buffer capability described elsewhere is also an essential element in the complete conforming process. The ESIG-4000 Enhanced Instancing capability (described later) is made possible by this dynamic conforming process.

Not only does hardware run-time conforming dramatically reduce modeling time, it also makes possible a significant reduction in the amount of storage required for databases. A single version of a feature can be used in just about any circumstance—the run-time conforming process takes care of making the model "fit" on the eventual terrain, regardless of terrain slope or terrain boundaries. This eliminates the need to create many versions of a feature to accommodate differing possible terrain-fit conditions.

A robust set of three conformality modes, described next, is available in the ESIG-4000. Each is suited to conforming a different type of feature onto the terrain.

Origin-Conformal Mode

Many features are designed to be placed on the terrain with the feature's origin "anchored" to (or offset from) the terrain. In the upper part of the next illustration, the tree is anchored at its base, and the house is modeled with respect to its center at ground level (with foundation extending below). The modeler creates the feature without concern for the (unknown) terrain shape. The feature is then "placed" relative to other features on a simple flat plane, with latitude, longitude locations obtained from DMA data, maps, photos, human input, etc.

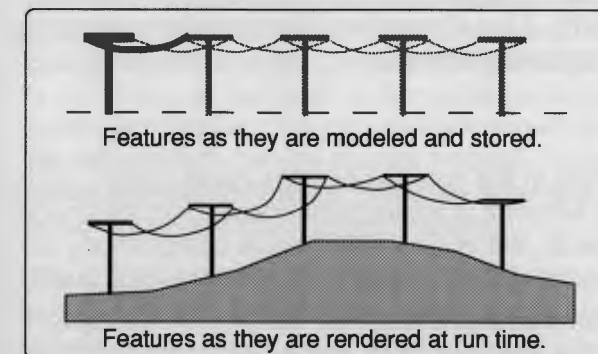


In the lower part of the illustration, the effect of the hardware's run-time conforming operation is seen, where the origin of each origin-conformal feature is adjusted to "ride" on the terrain at its proper vertical position. Optionally, an offset from the terrain can be specified. Note also that the R-Buffer takes care of any trimming of the features necessary to make them smoothly blend with the terrain surface.

Interestingly, this mode may also be used to simplify host-control of vehicles, where the host controls the latitude, longitude position, leaving Z to the hardware. For example, an aircraft could be set to maintain a 200 foot altitude above the terrain. The host need no longer repeatedly request height-above-terrain, and then compute Z from this result—only latitude, longitude need be controlled.

Line-Conformal Mode

Line-Conformal mode uses a line segment between two points on the terrain as the reference for placement of a linear feature (like telephone wires), rather than a point on the terrain. The vertices for the feature are individually conformed to points along the reference line.



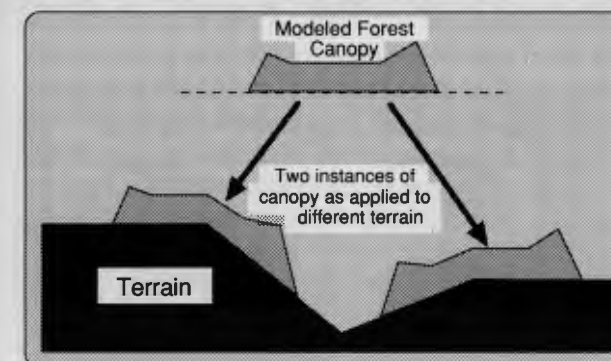
In this illustration a single origin-conformal telephone pole and a single line-conformal telephone wire are modeled. The ESIG-4000 creates multiple instances of the pole and wire, placing these at locations designated by the modeler (or the modeling tools from DFAD), again requiring him only to "place" the poles at their proper latitude, longitude locations on a simple flat plane. He doesn't care about the lay of the land or the details of connecting the wires.

It is particularly interesting to note in this case what happens as the poles are put into their correct positions by the ESIG-4000 hardware at run-time. Each instance of the wire is vertically adjusted to meet the crossbars on the poles.

The combination of several key ESIG-4000 features—origin conformal mode, line conformal mode, and Enhanced Instancing—are all brought together in this example. Not only are the poles placed at the correct positions, with the correct elevation values for the varying terrain, but the poles are "rotated" so that the crossbars are perpendicular to the path of the wires. In addition, the wires are scaled to allow their modeled curvature to properly meet the crossbars on the poles. The placement of the wires is referenced to the imaginary line segment joining any two poles, which is why this mode is referred to as "line-conformal."

Surface-Conformal Mode

This mode is similar to Line-Conformal mode, except that an *areal* feature (such as a road, pipeline, forest canopy, or airport runway) is processed to maintain a distance relative to the terrain surface.



In this illustration, a three-dimensional forest canopy section is created (two-dimensional cross section shown). The canopy is simply modeled upon an imaginary flat-plane database. Multiple instances of this single canopy model can then be placed in the database as dictated by the source data.

When the canopy is combined with the terrain at run-time, the canopy polygons are subdivided by the hardware at terrain polygon boundaries. Each resultant polygon is then adjusted according to the slope of the terrain polygon under it, causing the canopy to inherit the proper local slope.

Again, neither the modeler nor the modeling tools require any knowledge of the terrain slope, and need only place the features (canopy) where they belong.

Range Buffer for Generalized Visual Priority

Background

For about a decade the predominant method for determining visual priority in high-performance image generators has been the cellular-priority technique introduced with the E&S CT-5. This technique has provided excellent results for most simulation requirements. However, as database complexity continues to increase, and low-level flight with complex multi-vehicle simulation is required, new approaches must be employed which don't rely on modeled separating planes to determine visual priority. Also, elimination of the need for separating planes avoids one of the major time-consuming elements of database design.

The R-Buffer — An Enhancement to the Z-Buffer

The ESIG-4000 employs a full sub-pixel Range Buffer (R-Buffer) in the determination of visual priority. The R-Buffer is an enhancement of the classical Z-buffer approach, where Z represents the "depth" into the image (the dimension "away from" the viewer) of a given scene element. With ESIG-4000, range comparisons are made for each sample, with many samples being computed for each displayed pixel. This provides high quality anti-aliased polygon edges, along with the full benefits of dynamic occultation, including inter-penetration of objects.

One significant advantage of the R-Buffer over a conventional Z-buffer is the generalized support for fields of view greater than 180 degrees. Conventional Z-buffer approaches typically define positive Z in front of the eyepoint, and negative Z behind the eyepoint. Such systems have difficulty dealing with the small differences of Z near the XY plane (when Z approaches 0) often leading to priority anomalies near the XY plane. The R-Buffer is based on the *range* to the scene element, rather than only its Z value, providing usable results over the complete field of regard, even to 360 degrees.

As an added benefit, processing dynamic model occultation in the hardware frees up other system resources, permitting a larger number of models to be processed.

Role of the R-Buffer in the Separation of Terrain and Feature Processing

The separation of terrain and features in the ESIG-4000 relies on the dynamic occultation provided by the R-Buffer. As features are "conformed" to the terrain during run-time processing, the hardware takes care of giving each feature the correct position, scale, and orientation, along with the proper elevation for its particular terrain. The R-Buffer is the final step in the feature placement process, and is responsible for "trimming" the feature to mate properly with the terrain slope.

The following figure compares the separating plane implementation of feature placement, with the R-Buffer method. In the illustrations on the left, a house is shown, placed on two different terrain slopes. Note that the house fits properly on flat terrain, but leaves a gap on sloping terrain. When designing the house for an R-Buffer system, the house is made with enough "foundation" to accommodate the expected variation in terrain slope. When the house is placed on the terrain at run-time, the R-Buffer renders the proper intersection of the terrain surface with the house.



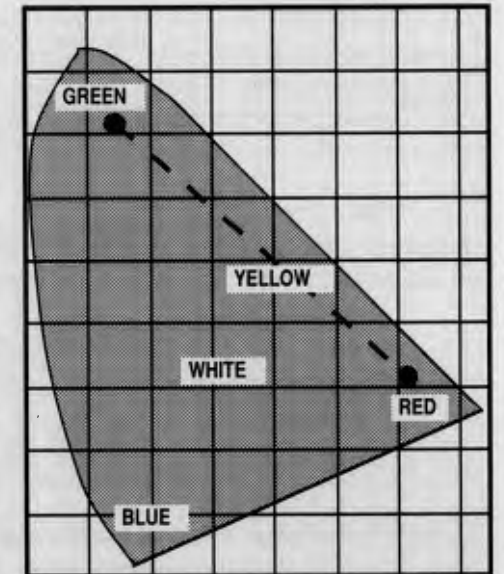
Increased Capacity for Moving Models

Since the R-Buffer inherently computes proper visual priority for moving models, without requiring special processing, moving models are not limited by occultation calculations in either hardware or software. With other schemes the capacity for moving models is often significantly limited by these calculations.

"Global Texture" Full Color Terrain Texture

A common method of texture processing inherently imposes a "two-color" limitation—the colors achievable from a single texture map are those which can be obtained by linear interpolation between a pair of RGB colors.

This illustration shows a standard CIE Chromaticity Diagram. All the colors achievable using combinations of the colors red, green, and blue are contained within the shaded area. An example of the two-color limitation is shown, where one color is green and the other color is red. Thus, all the texels in that particular texture map must be some linear combination of green and red, shown by the dashed line, such as yellow which the line passes through. No red/green combination could ever achieve white, blue, black, or any other color except the linear combinations of green and red.



Some areas *can* be adequately rendered with two-color texture, but many can not. Often the most important navigational and tactical features are multi-colored, such as cities and other man-made features. With two-color texture, these areas must be augmented with extra polygons for items varying in color from the terrain polygon's two-color motif. Such additional polygons generally don't convey *geometric* information, but only permit color variation, needlessly wasting polygon capacity.

Other techniques are available which provide some additional color flexibility, but still generally limit color freedom and/or suffer from undesirable aliasing artifacts or "blotchy" color variations. ESIG-4000's Global Texture provides full *three-color* terrain texture, removing the color constraints to database realism.

A Global Seamless Multi-Level Texture Space

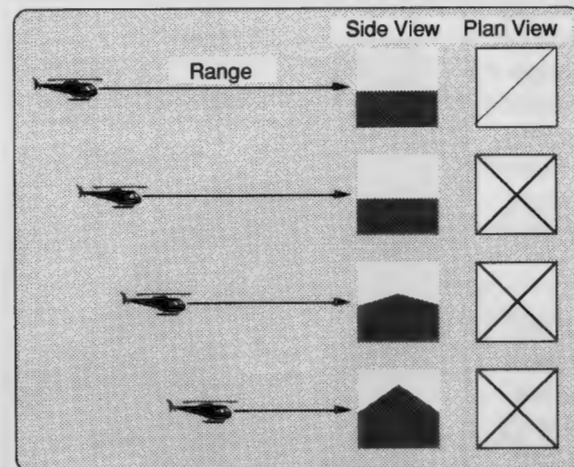
Global Texture provides the capability to apply "seamless" full color photo texture information of multiple resolutions to the terrain skin over an entire database. These multiple texture resolutions are blended to provide proper spatial frequency content over a very wide range of viewing distances.

One large-scale texture map—perhaps from a satellite photo several hundred miles on a side—can cover an entire database. While this large-scale photo provides good realism for distant views, it obviously cannot provide adequate detail for closer views. Global Texture allows many levels of detail, each successive level having a smaller scale, and increased resolution. At run time, texture information is obtained from the map level with proper resolution for the current viewing distance. Interpolations are performed between neighboring texels to avoid discrete changes in texture data (i.e. a "sugar cube" appearance). Interpolation also occurs when transitioning from one map level to the next. The result is a graceful blending of adjacent texels in a texture map, along with a smooth blend between map levels. The hardware uses the highest texture LOD which can be represented properly in the image, avoiding distracting aliasing effects.

Note that while the ESIG-4000 *permits* high texture LOD everywhere, it is only *required* where it is necessary for purposes of the mission, such as in target areas, or navigation waypoints. The system allows high-detail texture to be included where it *is* needed, but does not require it where it is *not* needed.

“Betweening” Continuous LOD Emergence

The ESIG-4000 employs a new technique for managing LOD transitions called “betweening.” This scheme is illustrated in the following figure, showing a section of terrain as it proceeds through one particular LOD transition.



In the top frame the terrain section is shown at the initial (lower) LOD—two polygons.

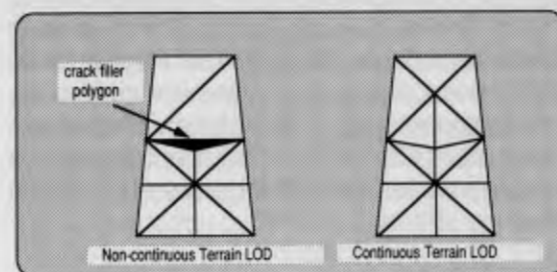
The second frame shows the first step of the betweening process. When the transition range is first reached, a new vertex is introduced. Note that the vertex is created *in the plane of* the existing low-LOD polygon. Thus the observer sees no change as the transition begins, because the resultant geometry is not altered.

In the third frame the aircraft has proceeded partway through the “transition zone,” resulting in the new vertex’s position partway “between” its initial and final positions.

As flight continues, the new vertex eventually reaches its final position, and the betweening process for that particular LOD transition is completed. Note that the above illustration exaggerates the amount of vertex travel in order to illustrate the concept of betweening. In practice, the objective is to carefully coordinate the LOD transition ranges with the amount of vertex travel during a betweening operation such that the vertex’s perceived *angular* travel is approximately constant, and small enough not to be obvious.

There are several very important benefits to the betweening process. First, since the process is continuous, there are no discrete events to attract the observer’s attention. This makes it possible to locate these LOD transitions at a closer range to the eyepoint. Hence terrain and features can proceed toward lower LOD more quickly with increasing range, resulting in a significant polygon savings.

Secondly, since the process is continuous, there is no possibility that “cracks” will develop in the terrain skin. With the discrete approach, there can be, and generally is, a geometric gap between adjoining LOD’s as shown in the next figure.



In this illustration the high LOD terrain has a lower elevation along the edge where it adjoins the low LOD terrain. This results in a terrain crack, which must be filled with the addition of a “crack-filler” polygon. This, unfortunately, adds complexity and time to the database creation process. With “betweening” (shown on the right), the LOD transition never creates such a discontinuity, obviating the need for crack-filler polygons.

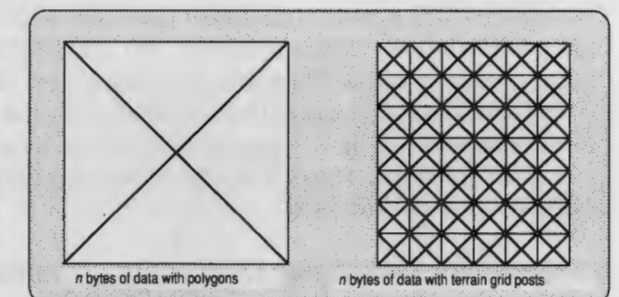
High Terrain Accuracy Gridded Terrain

Databases continue to grow, placing ever-increasing demands on visual systems. Multi-gigabyte databases are becoming routine, and there seems to be little danger of exceeding pilot demand for increased realism. To support this simultaneous drive for larger geographic areas, greater terrain fidelity, and higher feature density, effective techniques for data compression must be employed.

The ESIG-4000 stores terrain elevation data as a regular array of elevation values, rather than as explicit terrain polygons, providing a dramatic decrease in the amount of data which must be stored to describe the terrain skin. Storage of polygons can be relatively expensive, since each polygon generally needs the geometry (XYZ) for three vertices, shading information, intensity information, color information, and perhaps other miscellaneous data.

For storage of elevation grid posts, only the elevation (Z) value need be stored. The X and Y values are implied by the address of the Z elevation data. As a general rule, compression ratios of over 40:1 are achievable by storing grid-post data.

This figure exemplifies the difference between storage used for polygons, versus gridded elevation data. The left frame shows some terrain facet size, with terrain represented as polygons. On the right, the same storage contains gridded data, with about 40 times as much terrain data in that same storage, dramatically increasing terrain fidelity.



Improved Terrain Fidelity, Plus Faster Terrain Paging for High Speed Flight

This savings in terrain data relieves two crucial areas of IG storage—the database disks, and the hardware memories (EVM). Efficient utilization of these storage media is important both for increased geographic area, and increased terrain fidelity. For a given amount of storage, one can either represent a larger area, or provide increased terrain fidelity/accuracy, or both.

Terrain fidelity for low/fast flight simulation is often limited by the ability of the image generator to “page” terrain information from the database disks into the EVM. Using compressed gridded terrain data, paging bandwidth is effectively increased by the data compression ratio, resulting in correspondingly higher terrain fidelity/velocity.

In addition, this method of representing terrain is fundamentally compatible with available terrain data sources, such as DMA/DTED, Project 2851, and terrain data extracted from satellite photos. These data sources inherently produce regular grids of terrain elevation data, allowing the ESIG-4000 to eliminate the “polygon terrain skinning” step from the database modeling process. This allows more rapid generation of databases, since the hardware uses data essentially in its source form—an array of elevation values.

Terrain LOD is permitted to be heterogeneous, meaning that the terrain grid spacing can vary from area to area within a database, as dictated by terrain roughness, scenario requirements or availability of source data. The ESIG-4000 hardware supports terrain grid densities exceeding DMA/DTED Level 2.

Advanced Display Support: Non Linear Image Mapping

Projection of images onto non-planar surfaces requires distortion correction within the image, so that the resultant displayed image exhibits undistorted geometry. While some projection devices can perform such distortion, most do not, requiring this functionality to be performed by the image generator.

Wide Dynamic Range for Distortion Function — Variable Acuity Support

The advanced techniques for image generation employed in the ESIG-4000 can produce an image distortion with very generalized image mapping characteristics. The distortion correction function can exhibit a large dynamic range, suitable for dealing with a wide variety of different projection surfaces and display optics. In particular, the Non-Linear Image Mapping (NLIM) function can accommodate acute distortion curves such as might be required to drive a variable-acuity projector for use with eye-tracked systems.

Distortion Correction Standard in the Hardware

The ESIG-4000 system has been carefully engineered to anticipate the demanding requirements of future high-performance display systems. An important benefit of the resultant architecture is that these advanced image mapping capabilities are an integral part of the system. When non-linear image mapping is required, no additional hardware need be installed—only the software and data files for the particular projection geometry need be created. This is especially beneficial for systems where image mapping capability is to be added after system acquisition, where it would otherwise be necessary to undergo a field-upgrade to include extra-cost image mapping hardware.

No Capacity Penalty or Distracting Anomalies

The distortion-correction algorithms used in the ESIG-4000 have been engineered in such a way as to provide a generalized mapping capability with sufficient flexibility to meet a wide variety of distortion and image-mapping needs, all without polygon processing penalty, and without generating any “cracks” which disturb the imagery or need to be filled with “crack filler” polygons.

Compatible with Dynamic NLIM Option

The design is also compatible with a future option to provide dynamic NLIM capability, which is required for situations where the projector direction or location is dynamic, such that the geometry of intersection with the display surface changes dynamically (e.g. a head/eye tracked projector, located away from the center of the dome).

Improved Realism: Enhanced Features Instancing

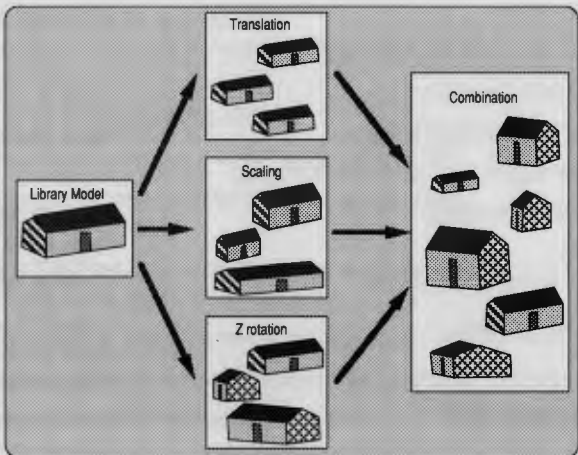
The term “instancing” refers to the use of a single library feature multiple times within a database. Typical examples are houses, trees, aircraft and other vehicles. These features are often used as the library items to satisfy references for features called out in DMA/DFAD descriptors, since the particular polygon geometry of the specific house/tree/etc. is not given in the DFAD data.

Traditional Limitations

Previous to ESIG-4000, the instancing function only provided the ability to locate (translate) the feature at the desired place. While this provided a powerful capability, significant additional effort was required if a more complete adherence to the DFAD/real-world item was required. For example, if the correct orientation was important, then a separate library feature had to be created for each unique orientation. Similarly, if the correct size was required, a separate version of the library feature was required for each unique size. This often led to either a large number of variations of feature models being created (at significant labor/CPU cost and storage penalty), or to significant compromises being made in database fidelity to the real world.

Enhanced Capabilities: Orientation and Scaling

With ESIG-4000, the capability exists to not only translate features to their desired locations, but to provide proper orientation and scale as well. Features can also be mirrored. This figure illustrates a library house model (modeled once, and stored once in the image generator), which can be used many times, each with a separate definition of location, orientation, and scale, as required to correlate with the source data.



Increased Realism, Plus Database Modeling Efficiency

Prior to this enhanced instancing capability, databases often suffered degraded real-world fidelity because of instancing limitations. Image generator memories have insufficient capacity to contain every orientation and size of every feature in the database, so these parameters were generally compromised during the process of “populating” a database with library features. The ESIG-4000 now permits each instance of a library feature to have the correct size and orientation, removing this constraint to database realism. In addition, this realism is available without having to create many separate versions of each model, yielding a significant reduction in database effort.

ESIG-4000 PERFORMANCE SUMMARY

PRIMARY PRODUCT EMPHASIS	<ul style="list-style-type: none">• Rapid database generation for mission rehearsal requirements• Efficient database production to reduce costs of frequent database creation• Highest in performance and capacity
MODES	<ul style="list-style-type: none">• Day/Dusk/Night/NVG (Night Vision Goggles)/FLIR (Forward Looking Infrared)
CHANNELS	<ul style="list-style-type: none">• Up to 8 per Environment Processor
VIEWPOINTS	<ul style="list-style-type: none">• Multiple viewpoints available, depending on specific application
VISUAL PRIORITY	<ul style="list-style-type: none">• Range Buffer (R-Buffer)<ul style="list-style-type: none">- Provides full occultation freedom for both moving and fixed objects- Supports interpenetration• Sub-pixel R-Buffer for high-quality anti-aliased polygon edges• Fixed-relationship priority within objects also supported• Proper occultation for all fields of view, including 180 degrees
TERRAIN/FEATURE SEPARATION	<ul style="list-style-type: none">• Separate terrain and feature databases permits very rapid database generation• Terrain stored as grid of elevation values, providing significant data compression• Terrain and features “conformed” (combined) in the hardware at run time• Flexible conforming modes, including:<ul style="list-style-type: none">- Origin conformal for “point” features such as trees and houses- Line conformal for “linear” features such as power lines- Surface conformal for features such as runways, pipelines, forest canopies
UPDATE/REFRESH RATE	<ul style="list-style-type: none">• 60 Hz or 50 Hz standard• Other rates including higher frequencies available
RESOLUTION	<ul style="list-style-type: none">• 250K to over 1.5M pixels per channel available with modular hardware configurations
DISPLAYED SURFACES (60 HZ)	<ul style="list-style-type: none">• 1,500 to 10,000 available with alternate resolutions and/or hardware configurations
PER CHANNEL	
TOTAL FOR SYSTEM	<ul style="list-style-type: none">• Up to 24,000 per Environment Processor
DISPLAYED LIGHTS (60 HZ, 0 POLYGONS)	
PER CHANNEL	<ul style="list-style-type: none">• Up to 30,000; trade 3:1 with polygons• Up to 8,000 light strings per channel
TOTAL FOR SYSTEM	<ul style="list-style-type: none">• Up to 24,000 light strings per Environment Processor
DISPLAY FORMAT	<ul style="list-style-type: none">• Raster; 2:1 vertical interlace standard• Pixels can be packed into non-rectangular raster shapes for optimizing resolution• Non-interlace configurations available
SCAN STANDARDS	<ul style="list-style-type: none">• Software programmable/switchable• Genlock capable
DISPLAY DEVICES	<ul style="list-style-type: none">• Software programmable output to drive a variety of display devices:<ul style="list-style-type: none">- Variable acuity displays- Head-or eye-slaved displays- Direct view CRT monitors, CRT projectors, light valve projectors- Helmet mounted displays- Display devices with non-rectangular active display areas
DISTORTION CORRECTION	<ul style="list-style-type: none">• Non-Linear Image Mapping (NLIM) hardware standard<ul style="list-style-type: none">- Wide dynamic range to drive advanced displays such as variable acuity- Dynamic NLIM option available- Compatible with all FOVs, including $\geq 180^\circ$- No capacity or performance compromise- Dynamic or static linear off-axis correction available
COLORS	<ul style="list-style-type: none">• One 4,096- or two 2,048-entry color tables per channel to support wide dynamic range sensor applications; can reload during simulation• Two entries per polygon to serve simultaneous visual and sensor channels
FEATURE TEXTURE	<ul style="list-style-type: none">• Hardware is standard• Up to 384 high resolution maps (128x128)• Programmable map resolution, from 16x16 up to 512x512• 4 texture maps per polygon for special effects, contour cutouts, high frequency detail, etc.• MIPmap LOD implementation for superior anti-aliasing• Can also be combined with Global Texture on terrain for up to 9 levels of micro-detail• Full-color capability• Texture motion for animation/motion without using dynamic coordinate systems (for clouds, water, smoke, etc.)

GLOBAL TEXTURE

DATABASE MANAGEMENT

OVERLOAD TECHNIQUES

FEATURE INSTANCING

FEATURE SIGNATURES

MOVING OBJECTS

ANIMATION

TRANSPORT DELAY

SIMULATION FUNCTIONS

- COLLISION DETECTION
- HEIGHT ABOVE TERRAIN
- LINE-OF-SIGHT RANGING
- INTERVISIBILITY

LANDING LIGHTS

LIGHT PATTERNS

ATMOSPHERIC EFFECTS

SURFACE EFFECTS

SPECIAL EFFECTS

SENSOR SIMULATION

LIBRARIES

GENERAL PURPOSE COMPUTER

Bold type indicates significant new features in the ESIG-4000.

Note: Some specifications are dependent on overall system configuration and database design, resulting in tradeoffs to optimize IG performance for a specific application.

- Contour texture for complex shapes with high edge quality
- Modulation texture for “wallpaper” texturing

- **For texturing terrain topology, separate from Feature Texture**
- **Full-color texture**, for complete RGB freedom
- **Single address space** covering entire database for seamless texture
- **Up to 20 MIP levels**, for texel size down to fractions of a meter
- **Photo-derived texture** from aerial or satellite photos for coarser LODs
- **Synthetically generated texture** or terrestrial photos for finer LODs
- **Tri-linear blending** for smooth texture transitions between texels and LODs

- **Separate databases and LOD processing** for terrain and features, allows terrain LOD and feature LOD to be optimized independently
- **“Betweening”** for continuous LOD transitions, allows shorter LOD transition ranges, providing more efficient polygon usage and higher scene density
- Also permits fade LOD blending where desired
- **Ability to classify each feature as 1 of 16 feature types**, and control LOD of each type separately
- **Capability to process database multiple times** to optimize LOD for different FOVs
- Area-of-Interest (AOI) LOD management for efficient database management across AOI inset/background

- Dynamic level of detail range adjustment, channelized dynamic polygon perspective size rejection, field time extension, frame rate update
- **Separate LOD management of each of 16 feature types**

- Provides not only position offset, **but also scale (size) and orientation (heading)** for each instance of a library feature

- **Each instance of a feature can be individually tailored** to provide variation between instances of same feature (include/exclude for sub-features)
- **Each instance of an areal feature** (e.g. forest) can be “trimmed” to provide much improved fit to specified areal shape

- **Up to 4,096 addressable dynamic coordinate systems (DCSs)**

- Virtually unlimited object representations for animation via rapid cycling, without using DCSs

- **3 field times standard** for ownship (i.e. 50 msec at 60 Hz including display field)
- **Shorter times available** for special applications

- **CD, HAT, LOSR, and intervisibility functions** include an “exclusion mask” to specify classes of features, such as ground, water, vegetation, and smoke

- Reports interference of point(s) on ownship or other DCS with predefined objects in the database

- Reports height of host-defined points or DCSs above terrain surface or above database features

- Reports range from viewpoint to specific database features in field of view

- **Reports intervisibility between points in database, 360° coverage**

- Up to 8 uniform or directional representations per channel
- Steerable lobe searchlight

- Flashing
- Rotating
- Directional
- Strobing

- Clouds
- Fog/haze
- Ground fog
- Patchy fog
- Glare
- Scud
- Horizon glow
- Lightning
- Thunderstorm cell
- Wet runway
- Snow covered runway

- Flat shading
- Smooth shading
- Fixed shading
- Color blending
- Transparency
- Self-luminous surfaces

- **Mountains protruding through clouds**
- **Dynamic flares** (illuminating the terrain and features)
- Multiple moving targets
- Fire and smoke
- Weapons effects

- Electro-Optical Sensor Simulation
- Infrared sensor simulation

- Scene components
- Ground vehicles
- Aircraft
- Cultural features
- Natural features

- **Internal VME 68K-based**
- **Ethernet, HSD** and other interfaces available